

# On Multilabel Classification and Ranking with Partial Feedback

Claudio Gentile

Universita' dell'Insubria

`claudio.gentile@uninsubria.it`

Francesco Orabona

Toyota Technological Institute at Chicago

`francesco@orabona.com`

July 3, 2012

## Abstract

We present a novel multilabel/ranking algorithm working in partial information settings. The algorithm is based on 2nd-order descent methods, and relies on upper-confidence bounds to trade-off exploration and exploitation. We analyze this algorithm in a partial adversarial setting, where covariates can be adversarial, but multilabel probabilities are ruled by (generalized) linear models. We show  $O(T^{1/2} \log T)$  regret bounds, which improve in several ways on the existing results. We test the effectiveness of our upper-confidence scheme by contrasting against full-information baselines on real-world multilabel datasets, often obtaining comparable performance.

## 1 Introduction

Consider a book recommendation system. Given a customer's profile, the system recommends a few possible books to the user by means of, e.g., a limited number of banners placed at different positions on a webpage. The system's goal is to select books that the user likes and possibly purchases. Typical feedback in such systems is the actual action of the user or, in particular, what books he has bought/preferred, if any. The system cannot observe what would have been the user's actions had other books got recommended, or had the same book ads been placed in a different order within the webpage. Such problems are collectively referred to as learning with partial feedback. As opposed to the full information case, where the system (the learning algorithm) knows the outcome of each possible response (e.g., the user's action for each and every possible book recommendation placed in the largest banner ad), in the partial feedback setting, the system only observes the response to very limited options and, specifically, the option that was actually recommended. In this and many other

examples of this sort, it is reasonable to assume that recommended options are not given the same treatment by the system, e.g., large banners which are displayed on top of the page should somehow be more committing as a recommendation than smaller ones placed elsewhere. Moreover, it is often plausible to interpret the user feedback as a preference (if any) *restricted* to the displayed alternatives.

We consider instantiations of this problem in the multilabel and learning-to-rank settings. Learning proceeds in rounds, in each time step  $t$  the algorithm receives an instance  $\mathbf{x}_t$  and outputs an ordered subset  $\hat{Y}_t$  of labels from a finite set of possible labels  $[K] = \{1, 2, \dots, K\}$ . Restrictions might apply to the size of  $\hat{Y}_t$  (due, e.g., to the number of available slots in the webpage). The set  $\hat{Y}_t$  corresponds to the aforementioned recommendations, and is intended to approximate the true set of preferences associated with  $\mathbf{x}_t$ . However, the latter set is never observed. In its stead, the algorithm receives  $Y_t \cap \hat{Y}_t$ , where  $Y_t \subseteq [K]$  is a *noisy version* of the true set of user preferences on  $\mathbf{x}_t$ . When we are restricted to  $|\hat{Y}_t| = 1$  for all  $t$ , this becomes a multiclass classification problem with bandit feedback – see below.

**Related work.** This paper lies at the intersection between online learning with partial feedback and multilabel classification/ranking. Both fields include a substantial amount of work, so we can hardly do it justice here. We outline some of the main contributions in the two fields, with an emphasis on those we believe are the most related to this paper.

A well-known and standard tool of facing the problem of partial feedback in online learning is to trade off exploration and exploitation through upper confidence bounds [20]. In the so-called *bandit* setting with contextual information (sometimes called bandits with side information or bandits with covariates, e.g., [2, 6, 9, 5, 19], and references therein) an online algorithm receives at each time step a *context* (typically, in the form of a feature vector  $\mathbf{x}$ ) and is compelled to select an action (e.g., a label), whose goodness is quantified by a predefined loss function. Full information about the loss function is not available. The specifics of the interaction model determines which pieces of loss will be observed by the algorithm, e.g., the actual value of the loss on the chosen action, some information on more profitable directions on the action space, noisy versions thereof, etc. The overall goal is to compete against classes of functions that map contexts to (expected) losses in a regret sense, that is, to obtain *sublinear* cumulative regret bounds. For instance, [2, 6, 9, 1] work in a finite action space where the mappings context-to-loss for each action are linear (or generalized linear, as in [9]) functions of the features. They all obtain  $T^{1/2}$ -like regret bounds, where  $T$  is the time horizon. This is extended in [19], where the loss function is modeled as a sample from a Gaussian process over the joint context-action space. We are using a similar (generalized) linear modeling here. Linear multiclass classification problems with bandit feedback are considered in, e.g., [16, 5, 14], where either  $T^{2/3}$  or  $T^{1/2}$  or even logarithmic regret bounds are proven, depending on the noise model and the underlying loss functions.

All the above papers do not consider *structured* action spaces, where the learner is afforded to select *sets* of actions, which is more suitable to multilabel and ranking problems. Along these lines are the papers [13, 27, 18, 25, 24]. The general problem of online minimization of a submodular loss function under both full and bandit information without covariates is considered in [13], achieving a regret  $T^{2/3}$  in the

bandit case. In [27] the problem of online learning of assignments is considered, where at each round an algorithm is requested to assign positions (e.g., rankings) to sets of items (e.g., ads) with given constraints on the set of items that can be placed in each position. Their problem shares similar motivations as ours but, again, the bandit version of their algorithm does not explicitly take side information into account, and leads to a  $T^{2/3}$  regret bound. In [18] the aim is to learn a suitable ordering (an “ordered slate”) of the available actions. Among other things, the authors prove a  $T^{1/2}$  regret bound in the bandit setting with a multiplicative weight updating scheme. Yet, no contextual information is incorporated. In [25] the ability of selecting sets of actions is motivated by a problem of diverse retrieval in large document collections which are meant to live in a general metric space. In contrast to our paper, that approach does not lead to strong regret guarantees for specific (e.g., smooth) loss functions. [24] uses a simple linear model for the hidden utility function of users interacting with a web system and providing partial feedback in any form that allows the system to make significant progress in learning this function (this is called an  $\alpha$ -informative feedback by the authors). A regret bound of  $T^{1/2}$  is again provided that depends on the degree of informativeness of the feedback. It is experimentally argued that this feedback is typically made available by a user that clicks on relevant URLs out of a list presented by a search engine. Despite the neatness of the argument, no formal effort is put into relating this information to the context information at hand or to the way data are generated.

The literature on multilabel learning and learning to rank is overwhelming. The wide attention this literature attracts is often motivated by its web-search-engine or recommender-system applications, and many of the papers are experimental in nature. Relevant references include [28, 11, 8], along with references therein. Moreover, when dealing with multilabel, the typical assumption is full supervision, an important concern being modeling correlations among classes. In contrast to that, the specific setting we are considering here need not face such a modeling [8]. Other related references are [15, 10], where learning is by pairs of examples. Yet, these approaches need i.i.d. assumptions on the data, and typically deliver batch learning procedures. To summarize, whereas we are technically close to [2, 6, 5, 9, 1, 19], from a motivational standpoint we are perhaps closest to [27, 18, 24].

**Our results.** We investigate the multilabel and learning-to-rank problems in a partial feedback scenario with contextual information, where we assume a probabilistic linear model over the labels, although the contexts can be chosen by an adaptive adversary. We consider two families of loss functions, one is a cost-sensitive multilabel loss that generalizes the standard Hamming loss in several respects, the other is a kind of (unnormalized) ranking loss. In both cases, the learning algorithm is maintaining a (generalized) linear predictor for the probability that a given label occurs, the ranking being produced by upper confidence-corrected estimated probabilities. In such settings, we prove  $T^{1/2} \log T$  cumulative regret bounds, which are essentially optimal (up to log factors). A distinguishing feature of our user feedback model is that, unlike previous papers (e.g., [13, 27, 1, 19]), we are not assuming the algorithm is observing a noisy version of the risk function on the currently selected action. In fact, when a generalized linear model is adopted, the mapping context-to-risk turns out to be nonconvex in the parameter space. Furthermore, when operating on structured action spaces this more traditional form of bandit model does not seem appropriate to capture the typical user

preference feedback. Our approach is based on having the loss decouple from the label generating model, the user feedback being a noisy version of the gradient of a *surrogate* convex loss associated with the model itself. As a consequence, the algorithm is not directly dealing with the original loss when making exploration. Though the emphasis is on theoretical results, we also validate our algorithms on two real-world multilabel datasets w.r.t. a number of loss functions, showing good comparative performance against simple multilabel/ranking baselines that operate with full information.

## 2 Model and preliminaries

We consider a setting where the algorithm receives at time  $t$  the side information vector  $\mathbf{x}_t \in \mathbb{R}^d$ , is allowed to output at a (possibly ordered) subset  $\hat{Y}_t \subseteq [K]$  of the set of possible labels, then the subset of labels  $Y_t \subseteq [K]$  associated with  $\mathbf{x}_t$  is generated, and the algorithm gets as feedback  $\hat{Y}_t \cap Y_t$ . The loss suffered by the algorithm may take into account several things: the *distance* between  $Y_t$  and  $\hat{Y}_t$  (both viewed as sets), as well as the *cost* for playing  $\hat{Y}_t$ . The cost  $c(\hat{Y}_t)$  associated with  $\hat{Y}_t$  might be given by the sum of costs suffered on each class  $i \in \hat{Y}_t$ , where we possibly take into account the *order* in which  $i$  occurs within  $\hat{Y}_t$  (viewed as an ordered list of labels). Specifically, given constant  $a \in [0, 1]$  and costs  $c = \{c(i, s), i = 1, \dots, s, s \in [K]\}$ , such that  $1 \geq c(1, s) \geq c(2, s) \geq \dots \geq c(s, s) \geq 0$ , for all  $s \in [K]$ , we consider the loss function

$$\ell_{a,c}(Y_t, \hat{Y}_t) = a |Y_t \setminus \hat{Y}_t| + (1 - a) \sum_{i \in \hat{Y}_t \setminus Y_t} c(j_i, |\hat{Y}_t|),$$

where  $j_i$  is the position of class  $i$  in  $\hat{Y}_t$ , and  $c(j_i, \cdot)$  depends on  $\hat{Y}_t$  only through its size  $|\hat{Y}_t|$ . In the above, the first term accounts for the false negative mistakes, hence there is no specific ordering of labels therein. The second term collects the loss contribution provided by all false positive classes, taking into account through the costs  $c(j_i, |\hat{Y}_t|)$  the order in which labels occur in  $\hat{Y}_t$ . The constant  $a$  serves as weighting the relative importance of false positive vs. false negative mistakes<sup>1</sup>. As a specific example, suppose that  $K = 10$ , the costs  $c(i, s)$  are given by  $c(i, s) = (s - i + 1)/s, i = 1, \dots, s$ , the algorithm plays  $\hat{Y}_t = (4, 3, 6)$ , but  $Y_t$  is  $\{1, 3, 8\}$ . In this case,  $|Y_t \setminus \hat{Y}_t| = 2$ , and  $\sum_{i \in \hat{Y}_t \setminus Y_t} c(j_i, |\hat{Y}_t|) = 3/3 + 1/3$ , i.e., the cost for mistakingly playing class 4 in the top slot of  $\hat{Y}_t$  is more damaging than mistakingly playing class 6 in the third slot. In the special case when all costs are unitary, there is no longer need to view  $\hat{Y}_t$  as an ordered collection, and the above loss reduces to a standard Hamming-like loss between sets  $Y_t$  and  $\hat{Y}_t$ , i.e.,  $a |Y_t \setminus \hat{Y}_t| + (1 - a) |\hat{Y}_t \setminus Y_t|$ . Notice that the partial feedback  $\hat{Y}_t \cap Y_t$  allows the algorithm to know which of the chosen classes in  $\hat{Y}_t$  are good or bad (and to what extent, because of the selected ordering within  $\hat{Y}_t$ ). Yet, the algorithm does not observe the value of  $\ell_{a,c}(Y_t, \hat{Y}_t)$  because  $Y_t \setminus \hat{Y}_t$  remains hidden.

Working with the above loss function makes the algorithm's output  $\hat{Y}_t$  become a *ranked* list of classes, where ranking is restricted to the deemed relevant classes only. In our setting, only a relevance feedback among the selected classes is observed (the set  $Y_t \cap \hat{Y}_t$ ), but no supervised ranking information (e.g., in the form of pairwise preferences) is provided to the algorithm within this set. Alternatively, we can think of a

<sup>1</sup>Notice that  $a$  is not redundant here, since the costs  $c(i, s)$  have been normalized to  $[0, 1]$ .

ranking framework where restrictions on the size of  $\hat{Y}_t$  are set by an exogenous (and possibly time-varying) parameter of the problem, and the algorithm is required to provide a ranking complying with these restrictions. More on the connection to the ranking setting with partial feedback is in Sec. 4.

The problem arises as to which noise model we should adopt so as to encompass significant real-world settings while at the same time affording *efficient implementation* of the resulting algorithms. For any subset  $Y_t \subseteq [K]$ , we let  $(y_{1,t}, \dots, y_{K,t}) \in \{0, 1\}^K$  be the corresponding indicator vector. Then it is easy to see that  $\ell_{a,c}(Y_t, \hat{Y}_t) = a \sum_{i=1}^K y_{i,t} + (1-a) \sum_{i \in \hat{Y}_t} \left( c(j_i, |\hat{Y}_t|) - \left( \frac{a}{1-a} + c(j_i, |\hat{Y}_t|) \right) y_{i,t} \right)$ . Moreover, because the first sum does not depend on  $\hat{Y}_t$ , for the sake of optimizing over  $\hat{Y}_t$  we can equivalently define

$$\ell_{a,c}(Y_t, \hat{Y}_t) = (1-a) \sum_{i \in \hat{Y}_t} \left( c(j_i, |\hat{Y}_t|) - \left( \frac{a}{1-a} + c(j_i, |\hat{Y}_t|) \right) y_{i,t} \right). \quad (1)$$

Let  $\mathbb{P}_t(\cdot)$  be a shorthand for the conditional probability  $\mathbb{P}_t(\cdot | \mathbf{x}_t)$ , where the side information vector  $\mathbf{x}_t$  can in principle be generated by an adaptive adversary as a function of the past. Then  $\mathbb{P}_t(y_{1,t}, \dots, y_{K,t}) = \mathbb{P}(y_{1,t}, \dots, y_{K,t} | \mathbf{x}_t)$ , where the marginals  $\mathbb{P}_t(y_{i,t} = 1)$  satisfy<sup>2</sup>

$$\mathbb{P}_t(y_{i,t} = 1) = \frac{g(-\mathbf{u}_i^\top \mathbf{x}_t)}{g(\mathbf{u}_i^\top \mathbf{x}_t) + g(-\mathbf{u}_i^\top \mathbf{x}_t)}, \quad i = 1, \dots, K, \quad (2)$$

for some  $K$  vectors  $\mathbf{u}_1, \dots, \mathbf{u}_K \in \mathcal{R}^d$  and some (known) function  $g : D \subseteq \mathcal{R} \rightarrow \mathcal{R}^+$ . The model is well defined if  $\mathbf{u}_i^\top \mathbf{x} \in D$  for all  $i$  and all  $\mathbf{x} \in \mathcal{R}^d$  chosen by the adversary. We assume for the sake of simplicity that  $\|\mathbf{x}_t\| = 1$  for all  $t$ . Notice that the variables  $y_{i,t}$  need not be conditionally independent. We are only defining a family of allowed joint distributions  $\mathbb{P}_t(y_{1,t}, \dots, y_{K,t})$  through the properties of their marginals  $\mathbb{P}_t(y_{i,t})$ .

The function  $g$  above will be instantiated to the negative derivative of a suitable convex and nonincreasing loss function  $L$  which our algorithm will be based upon. For instance, if  $L$  is the square loss  $L(\Delta) = (1 - \Delta)^2/2$ , then  $g(\Delta) = 1 - \Delta$ , resulting in  $\mathbb{P}_t(y_{i,t} = 1) = (1 + \mathbf{u}_i^\top \mathbf{x}_t)/2$ , under the assumption  $D = [-1, 1]$ . If  $L$  is the logistic loss  $L(\Delta) = \ln(1 + e^{-\Delta})$ , then  $g(\Delta) = (e^\Delta + 1)^{-1}$ , and  $\mathbb{P}_t(y_{i,t} = 1) = e^{\mathbf{u}_i^\top \mathbf{x}_t} / (e^{\mathbf{u}_i^\top \mathbf{x}_t} + 1)$ , with domain  $D = \mathcal{R}$ .

Set for brevity  $\Delta_{i,t} = \mathbf{u}_i^\top \mathbf{x}_t$ . Taking into account (1), this model allows us to write the (conditional) expected loss of the algorithm playing  $\hat{Y}_t$  as

$$\mathbb{E}_t[\ell_{a,c}(Y_t, \hat{Y}_t)] = \sum_{i \in \hat{Y}_t} \left( c(j_i, |\hat{Y}_t|) - \left( \frac{a}{1-a} + c(j_i, |\hat{Y}_t|) \right) p_{i,t} \right), \quad (3)$$

where  $p_{i,t} = \frac{g(-\Delta_{i,t})}{g(\Delta_{i,t}) + g(-\Delta_{i,t})}$ , and the expectation  $\mathbb{E}_t$  above is w.r.t. the generation of labels  $Y_t$ , conditioned on both  $\mathbf{x}_t$ , and all previous  $\mathbf{x}$  and  $Y$ . A key aspect of this formalization is that the Bayes optimal ordered subset  $Y_t^* = \operatorname{argmin}_{Y=(j_1, j_2, \dots, j_{|Y|}) \subseteq [K]} \mathbb{E}_t[\ell_{a,c}(Y_t, Y)]$

<sup>2</sup> The reader familiar with generalized linear models will recognize the derivative of the function  $p(\Delta) = \frac{g(-\Delta)}{g(\Delta) + g(-\Delta)}$  as the (inverse) link function of the associated canonical exponential family of distributions [21].

can be computed efficiently when knowing  $\Delta_{1,t}, \dots, \Delta_{K,t}$ . This is handled by the next lemma. In words, this lemma says that, in order to minimize (3), it suffices to try out all possible sizes  $s = 0, 1, \dots, K$  for  $Y_t^*$  and, for each such value, determine the sequence  $Y_{s,t}^*$  that minimizes (3) over all sequences of size  $s$ . In turn,  $Y_{s,t}^*$  can be computed just by sorting classes  $i \in [K]$  in decreasing order of  $p_{i,t}$ , sequence  $Y_{s,t}^*$  being given by the first  $s$  classes in this sorted list.<sup>3</sup>

**Lemma 1.** *With the notation introduced so far, let  $p_{i_1,t} \geq p_{i_2,t} \geq \dots p_{i_K,t}$  be the sequence of  $p_{i,t}$  sorted in nonincreasing order. Then we have that  $Y_t^* = \operatorname{argmin}_{s=0,1,\dots,K} \mathbb{E}_t[\ell_{a,c}(Y_t, Y_{s,t}^*)]$ , where  $Y_{s,t}^* = (i_1, i_2, \dots, i_s)$ , and  $Y_{0,t}^* = \emptyset$ .*

Notice the way costs  $c(i, s)$  influence the Bayes optimal computation. We see from (3) that placing class  $i$  within  $\hat{Y}_t$  in position  $j_i$  is beneficial (i.e., it leads to a reduction of loss) if and only if  $p_{i,t} > c(j_i, |\hat{Y}_t|) / (\frac{a}{1-a} + c(j_i, |\hat{Y}_t|))$ . Hence, the higher is the slot  $j_i$  in  $\hat{Y}_t$  the larger should be  $p_{i,t}$  in order for this inclusion to be convenient.<sup>4</sup> It is  $Y_t^*$  that we interpret as the true set of user preferences on  $\mathbf{x}_t$ .

We would like to compete against the above  $Y_t^*$  in a cumulative regret sense, i.e., we would like to bound  $R_T = \sum_{t=1}^T \mathbb{E}_t[\ell_{a,c}(Y_t, \hat{Y}_t)] - \mathbb{E}_t[\ell_{a,c}(Y_t, Y_t^*)]$  with high probability. Inspired by [5], we devise an online second-order descent algorithm whose updating rule makes the comparison vector  $U = (\mathbf{u}_1, \dots, \mathbf{u}_K) \in \mathcal{R}^{dK}$  defined through (2) be Bayes optimal w.r.t. a surrogate convex loss  $L(\cdot)$  such that  $g(\Delta) = -L'(\Delta)$ . Observe that the expected loss function (3) is, generally speaking, nonconvex in the margins  $\Delta_{i,t}$  (consider, for instance the logistic case  $g(\Delta) = \frac{1}{e^{\Delta} + 1}$ ). Thus, we cannot directly minimize this expected loss.

### 3 Algorithm and regret bounds

In Figure 1 is our bandit algorithm for (ordered) multiple labels. The algorithm is based on replacing the unknown model vectors  $\mathbf{u}_1, \dots, \mathbf{u}_K$  with prototype vectors  $\mathbf{w}'_{1,t}, \dots, \mathbf{w}'_{K,t}$ , being  $\mathbf{w}'_{i,t}$  the time- $t$  approximation to  $\mathbf{u}_i$ , satisfying similar constraints we set for the  $\mathbf{u}_i$  vectors. For the sake of brevity, we let  $\hat{\Delta}'_{i,t} = \mathbf{x}_t^\top \mathbf{w}'_{i,t}$ , and  $\Delta_{i,t} = \mathbf{u}_i^\top \mathbf{x}_t$ ,  $i \in [K]$ . The algorithm uses  $\hat{\Delta}'_{i,t}$  as proxies for the underlying  $\Delta_{i,t}$  according to the (upper confidence) approximation scheme  $\Delta_{i,t} \approx [\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D$ , where  $\epsilon_{i,t} \geq 0$  is a suitable upper-confidence level for class  $i$  at time  $t$ , and  $[\cdot]_D$  denotes the clipping-to- $D$  operation, i.e.,  $[x]_D = \max(\min(x, D), -D)$ . The algorithm's prediction at time  $t$  has the same form as the computation of the Bayes optimal sequence  $Y_t^*$ , where we replace the true (and unknown)  $p_{i,t} = \frac{g(-\Delta_{i,t})}{g(\Delta_{i,t}) + g(-\Delta_{i,t})}$  with the corresponding upper confidence proxy  $\hat{p}_{i,t} = \frac{g(-\hat{\Delta}'_{i,t} + \epsilon_{i,t})}{g([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) + g(-[\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D)}$ . Computing  $\hat{Y}_t$  can be done by mimicking the computation of the Bayes optimal  $Y_t^*$  (just replace

<sup>3</sup> Due to space limitations, all proofs are given in the supplementary material.

<sup>4</sup> Notice that this depends on the actual size of  $\hat{Y}_t$ , so we cannot decompose this problem into  $K$  independent problems. The decomposition does occur if the costs  $c(i, s)$  are constants, independent of  $i$  and  $s$ , and the criterion for inclusion becomes  $p_{i,t} \geq \theta$ , for some constant threshold  $\theta$ .

**Parameters:** loss parameters  $a \in [0, 1]$ , cost values  $c(i, s)$ , interval  $D = [-R, R]$ , function  $g : D \rightarrow \mathcal{R}$ , confidence level  $\delta \in [0, 1]$ .

**Initialization:**  $A_{i,0} = I \in \mathcal{R}^{d \times d}$ ,  $i = 1, \dots, K$ ,  $\mathbf{w}_{i,1} = \mathbf{0} \in \mathcal{R}^d$ ,  $i = 1, \dots, K$ ;

**For**  $t = 1, 2, \dots, T$  :

1. Get instance  $\mathbf{x}_t \in \mathcal{R}^d : \|\mathbf{x}_t\| = 1$ ;
2. For  $i \in [K]$ , set  $\hat{\Delta}'_{i,t} = \mathbf{x}_t^\top \mathbf{w}'_{i,t}$ , where

$$\mathbf{w}'_{i,t} = \begin{cases} \mathbf{w}_{i,t} & \text{if } \mathbf{w}_{i,t}^\top \mathbf{x}_t \in [-R, R], \\ \mathbf{w}_{i,t} - \left( \frac{\mathbf{w}_{i,t}^\top \mathbf{x}_t - R \operatorname{sign}(\mathbf{w}_{i,t}^\top \mathbf{x}_t)}{\mathbf{x}_t^\top A_{i,t-1}^{-1} \mathbf{x}_t} \right) A_{i,t-1}^{-1} \mathbf{x}_t & \text{otherwise;} \end{cases}$$

3. Output

$$\hat{Y}_t = \operatorname{argmin}_{Y=(j_1, j_2, \dots, j_{|Y|}) \subseteq [K]} \left( \sum_{i \in Y} \left( c(j_i, |Y|) - \left( \frac{a}{1-a} + c(j_i, |Y|) \right) \hat{p}_{i,t} \right) \right),$$

$$\text{where: } \hat{p}_{i,t} = \frac{g(-[\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D)}{g([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) + g(-[\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D)},$$

$$\epsilon_{i,t}^2 = \mathbf{x}_t^\top A_{i,t-1}^{-1} \mathbf{x}_t \left( U^2 + \frac{d c'_L}{(c'_L)^2} \ln \left( 1 + \frac{t-1}{d} \right) + \frac{12}{c'_L} \left( \frac{c'_L}{c'_L} + 3L(-R) \right) \ln \frac{K(t+4)}{\delta} \right);$$

4. Get feedback  $Y_t \cap \hat{Y}_t$ ;

5. For  $i \in [K]$ , update  $A_{i,t} = A_{i,t-1} + |s_{i,t}| \mathbf{x}_t \mathbf{x}_t^\top$ ,  $\mathbf{w}_{i,t+1} = \mathbf{w}'_{i,t} - \frac{1}{c'_L} A_{i,t}^{-1} \nabla_{i,t}$ ,

where

$$s_{i,t} = \begin{cases} 1 & \text{If } i \in Y_t \cap \hat{Y}_t \\ -1 & \text{If } i \in \hat{Y}_t \setminus Y_t = \hat{Y}_t \setminus (Y_t \cap \hat{Y}_t) \\ 0 & \text{otherwise;} \end{cases}$$

$$\text{and } \nabla_{i,t} = \nabla_{\mathbf{w}} L(s_{i,t} \mathbf{w}^\top \mathbf{x}_t)|_{\mathbf{w}=\mathbf{w}'_{i,t}} = -g(s_{i,t} \hat{\Delta}'_{i,t}) s_{i,t} \mathbf{x}_t.$$

Figure 1: The partial feedback algorithm in the (ordered) multiple label setting.

$p_{i,t}$  by  $\hat{p}_{i,t}$ , i.e., order of  $K \log K$  running time per prediction. Thus the algorithm is producing a ranked list of relevant classes based on upper-confidence-corrected scores  $\hat{p}_{i,t}$ . Class  $i$  is deemed relevant and ranked high among the relevant ones when either  $\hat{\Delta}'_{i,t}$  is a good approximation to  $\Delta_{i,t}$  and  $p_{i,t}$  is large, or when the algorithm is not very confident on its own approximation about  $i$  (that is, the upper confidence level  $\epsilon_{i,t}$  is large).

The algorithm receives in input the loss parameters  $a$  and  $c(i, s)$ , the model function  $g(\cdot)$  and the associated margin domain  $D = [-R, R]$ , and maintains both  $K$  positive definite matrices  $A_{i,t}$  of dimension  $d$  (initially set to the  $d \times d$  identity matrix), and  $K$  weight vector  $\mathbf{w}_{i,t} \in \mathcal{R}^d$  (initially set to the zero vector). At each time step  $t$ , upon receiving the  $d$ -dimensional instance vector  $\mathbf{x}_t$  the algorithm uses the weight vectors  $\mathbf{w}_{i,t}$  to compute the prediction vectors  $\mathbf{w}'_{i,t}$ . These vectors can easily be seen as the result of projecting  $\mathbf{w}_{i,t}$  onto interval  $D = [-R, R]$  w.r.t. the distance function  $d_{i,t-1}$ , i.e.,  $\mathbf{w}'_{i,t} = \operatorname{argmin}_{\mathbf{w} \in \mathcal{R}^d : \mathbf{w}^\top \mathbf{x}_t \in D} d_{i,t-1}(\mathbf{w}, \mathbf{w}_{i,t})$ ,  $i \in [K]$ , where  $d_{i,t}(\mathbf{u}, \mathbf{w}) = (\mathbf{u} - \mathbf{w})^\top A_{i,t} (\mathbf{u} - \mathbf{w})$ . Vectors  $\mathbf{w}'_{i,t}$  are then used to produce prediction values  $\hat{\Delta}'_{i,t}$  involved in the upper-confidence calculation of  $\hat{Y}_t \subseteq [K]$ . Next, the feedback  $Y_t \cap \hat{Y}_t$  is observed, and the algorithm in Figure 1 promotes all classes  $i \in Y_t \cap \hat{Y}_t$  (sign  $s_{i,t} = 1$ ), demotes all classes  $i \in \hat{Y}_t \setminus Y_t$  (sign  $s_{i,t} = -1$ ), and leaves

all remaining classes  $i \notin \hat{Y}_t$  unchanged (sign  $s_{i,t} = 0$ ). The update  $\mathbf{w}'_{i,t} \rightarrow \mathbf{w}_{i,t+1}$  is based on the gradients  $\nabla_{i,t}$  of a loss function  $L(\cdot)$  satisfying  $L'(\Delta) = -g(\Delta)$ . On the other hand, the update  $A_{i,t-1} \rightarrow A_{i,t}$  uses the rank one matrix  $\mathbf{x}_t \mathbf{x}_t^\top$ . In both the update of  $\mathbf{w}'_{i,t}$  and the one involving  $A_{i,t-1}$ , the reader should observe the role played by the signs  $s_{i,t}$ . Finally, the constants  $c'_L$  and  $c''_L$  occurring in the expression for  $\epsilon_{i,t}^2$  are related to smoothness properties of  $L(\cdot)$  – see next theorem.

**Theorem 2.** *Let  $L : D = [-R, R] \subseteq \mathcal{R} \rightarrow \mathcal{R}^+$  be a  $C^2(D)$  convex and non-increasing function of its argument,  $(\mathbf{u}_1, \dots, \mathbf{u}_K) \in \mathcal{R}^{dK}$  be defined in (2) with  $g(\Delta) = -L'(\Delta)$  for all  $\Delta \in D$ , and such that  $\|\mathbf{u}_i\| \leq U$  for all  $i \in [K]$ . Assume there are positive constants  $c_L$ ,  $c'_L$  and  $c''_L$  such that: i.  $\frac{L'(\Delta)L''(-\Delta)+L''(\Delta)L'(-\Delta)}{(L'(\Delta)+L'(-\Delta))^2} \geq -c_L$  and ii.  $(L'(\Delta))^2 \leq c'_L$ , and iii.  $L''(\Delta) \geq c''_L$  hold for all  $\Delta \in D$ . Then the cumulative regret  $R_T$  of the algorithm in Figure 1 satisfies, with probability at least  $1 - \delta$ ,*

$$R_T = O\left((1-a)c_L K \sqrt{TCd \ln\left(1 + \frac{T}{d}\right)}\right),$$

$$\text{where } C = O\left(U^2 + \frac{dc'_L}{(c''_L)^2} \ln\left(1 + \frac{T}{d}\right) + \left(\frac{c'_L}{(c''_L)^2} + \frac{L(-R)}{c''_L}\right) \ln \frac{KT}{\delta}\right).$$

It is easy to see that when  $L(\cdot)$  is the square loss  $L(\Delta) = (1 - \Delta)^2/2$  and  $D = [-1, 1]$ , we have  $c_L = 1/2$ ,  $c'_L = 4$  and  $c''_L = 1$ ; when  $L(\cdot)$  is the logistic loss  $L(\Delta) = \ln(1 + e^{-\Delta})$  and  $D = [-R, R]$ , we have  $c_L = 1/4$ ,  $c'_L \leq 1$  and  $c''_L = \frac{1}{2(1+\cosh(R))}$ , where  $\cosh(x) = \frac{e^x + e^{-x}}{2}$ .

**Remark 1.** A drawback of Theorem 2 is that, in order to properly set the upper confidence levels  $\epsilon_{i,t}$ , we assume prior knowledge of the norm upper bound  $U$ . Because this information is often unavailable, we present here a simple modification to the algorithm that copes with this limitation. We change the definition of  $\epsilon_{i,t}^2$  in Figure 1 to  $\epsilon_{i,t}^2 =$

$$\max\left\{\mathbf{x}^\top A_{i,t-1}^{-1} \mathbf{x} \left(\frac{2dc'_L}{(c''_L)^2} \ln\left(1 + \frac{t-1}{d}\right) + \frac{12}{c''_L} \left(\frac{c'_L}{c''_L} + 3L(-R)\right) \ln \frac{K(t+4)}{\delta}\right), 4R^2\right\}. \text{ This immediately leads to the following result.}$$

**Theorem 3.** *With the same assumptions and notation as in Theorem 2, if we replace  $\epsilon_{i,t}^2$  as explained above we have that, with probability at least  $1 - \delta$ ,  $R_T$  satisfies*

$$R_T = O\left((1-a)c_L K \sqrt{TCd \ln\left(1 + \frac{T}{d}\right)} + (1-a)c_L K R d \left(\exp\left(\frac{(c'_L)^2 U^2}{c''_L d}\right) - 1\right)\right).$$

## 4 On ranking with partial feedback

As Lemma 1 points out, when the cost values  $c(i, s)$  in  $\ell_{a,c}$  are strictly decreasing then the Bayes optimal ordered sequence  $Y_t^*$  on  $\mathbf{x}_t$  can be obtained by sorting classes in decreasing values of  $p_{i,t}$ , and then decide on a cutoff point<sup>5</sup> induced by the loss parameters, so as to tell relevant classes apart from irrelevant ones. In turn, because

<sup>5</sup> This is called the *zero point* in [11].



$p(\Delta) = \frac{g(-\Delta)}{g(\Delta)+g(-\Delta)}$  is increasing in  $\Delta$ , this ordering corresponds to sorting classes in decreasing values of  $\Delta_{i,t}$ . Now, if parameter  $a$  in  $\ell_{a,c}$  is very close<sup>6</sup> to 1, then  $|Y_t^*| = K$ , and the algorithm itself will produce ordered subsets  $\hat{Y}_t$  such that  $|\hat{Y}_t| = K$ . The resulting algorithm can thus be  $\Delta_{i_1,t} \geq \Delta_{i_2,t} \geq \dots \geq \Delta_{i_K,t}$  over all classes. Moreover, it does so by receiving *full* feedback on the relevant classes at time  $t$  (since  $Y_t \cap \hat{Y}_t = Y_t$ ). As is customary (e.g., [8]), one can view any multilabel assignment  $Y = (y_1, \dots, y_K) \in \{0, 1\}^K$  as a ranking among the  $K$  classes in the most natural way:  $i$  precedes  $j$  if and only if  $y_i > y_j$ . The (unnormalized) ranking loss function  $\ell_{rank}(Y, \hat{f})$  between the multilabel  $Y$  and a ranking function  $\hat{f} : \mathcal{R}^d \rightarrow \mathcal{R}^K$ , representing degrees of class relevance sorted in a decreasing order  $\hat{f}_{j_1}(\mathbf{x}_t) \geq \hat{f}_{j_2}(\mathbf{x}_t) \geq \dots \geq \hat{f}_{j_K}(\mathbf{x}_t)$ , counts the number of class pairs that disagree in the two rankings:  $\ell_{rank}(Y, \hat{f}) = \sum_{i,j \in [K]: y_i > y_j} \left( \{ \hat{f}_i(\mathbf{x}_t) < \hat{f}_j(\mathbf{x}_t) \} + \frac{1}{2} \{ \hat{f}_i(\mathbf{x}_t) = \hat{f}_j(\mathbf{x}_t) \} \right)$ , where  $\{\dots\}$  is the indicator function of the predicate at argument. As pointed out in [8], the ranking function  $\hat{f}(\mathbf{x}_t) = (p_{1,t}, \dots, p_{K,t})$  is also Bayes optimal w.r.t.  $\ell_{rank}(Y, \hat{f})$ , *no matter if* the class labels  $y_i$  are conditionally independent or not. Hence we can use this algorithm for tackling ranking problems derived from multilabel ones, when the measure of choice is  $\ell_{rank}$  and the feedback is full.

In fact, a partial information version of the above can easily be obtained. Suppose that at each time  $t$ , the environment discloses both  $\mathbf{x}_t$  and a maximal *size*  $S_t$  for the ordered subset  $\hat{Y}_t = (j_1, j_2, \dots, j_{|\hat{Y}_t|})$  (both  $\mathbf{x}_t$  and  $S_t$  can be chosen adaptively by an adversary). Here  $S_t$  might be the number of available slots in a webpage or the number of URLs returned by a search engine in response to query  $\mathbf{x}_t$ . Then it is plausible to compete in a regret sense against the best time- $t$  offline ranking of the form  $f(\mathbf{x}_t) = (f_1(\mathbf{x}_t), f_2(\mathbf{x}_t), \dots, f_h(\mathbf{x}_t), 0, \dots, 0)$ , with  $h \leq S_t$ . Further, the ranking loss could be reasonably restricted to count the number of class pairs disagreeing within  $\hat{Y}_t$  plus the number of false negative mistakes. E.g., if  $\hat{f}_{j_1}(\mathbf{x}_t) \geq \hat{f}_{j_2}(\mathbf{x}_t) \geq \dots \geq \hat{f}_{j_{|\hat{Y}_t|}}(\mathbf{x}_t)$ , we can set

$$\ell_{rank,t}(Y, \hat{f}) = \sum_{i,j \in \hat{Y}_t: y_i > y_j} \left( \{ \hat{f}_i(\mathbf{x}_t) < \hat{f}_j(\mathbf{x}_t) \} + \frac{1}{2} \{ \hat{f}_i(\mathbf{x}_t) = \hat{f}_j(\mathbf{x}_t) \} \right) + |Y_t \setminus \hat{Y}_t|.$$

It is not hard to see that the Bayes optimal ranking for  $\ell_{rank,t}$  is given by  $f^*(\mathbf{x}_t; S_t) = (p_{i_1,t}, \dots, p_{i_{S_t},t}, 0, \dots, 0)$ . If we put on the argmin (Step 3 in Figure 1) the further constraint  $|Y| \leq S_t$  (notice that the computation is still about sorting classes according to decreasing values of  $\hat{p}_{i,t}$ ), one can prove the following ranking counterpart to Theorem 2.

**Theorem 4.** *With the same assumptions and notation as in Theorem 2, let the cumulative regret  $R_T$  w.r.t.  $\ell_{rank,t}$  be defined as*

$$R_T = \sum_{t=1}^T \mathbb{E}_t[\ell_{rank,t}(Y_t, (\hat{p}_{j_1,t}, \dots, \hat{p}_{j_{S_t},t}, 0, \dots, 0))] - \mathbb{E}_t[\ell_{rank,t}(Y_t, (p_{i_1,t}, \dots, p_{i_{S_t},t}, 0, \dots, 0))],$$

<sup>6</sup> If  $a = 1$ , the algorithm only cares about false negative mistakes, the best strategy being always predicting  $\hat{Y}_t = [K]$ . Unsurprisingly, this yields zero regret in both Theorems 2 and 3.

where  $\hat{p}_{j_1,t} \geq \dots \geq \hat{p}_{j_{S_t},t} \geq 0$  and  $p_{i_1,t} \geq \dots \geq p_{i_{S_t},t} \geq 0$ . Then, with probability at least  $1 - \delta$ , we have  $R_T = O\left(c_L \sqrt{S K T C d \ln\left(1 + \frac{T}{d}\right)}\right)$ , where  $S = \max_{t=1,\dots,T} S_t$ .

The proof (see the appendix) is very similar to the one of Theorem 2. This suggests that, to some extent, we are decoupling the label generating model from the loss function  $\ell$  under consideration. Notice that the linear dependence on the total number of classes  $K$  (which is often much larger than  $S$  in a multilabel/ranking problem) is replaced by  $\sqrt{SK}$ . One could get similar benefits out of Theorem 2. Finally, one could also combine Theorem 4 with the argument contained in Remark 1.

## 5 Experiments and conclusions

The experiments we report here are meant to validate the exploration-exploitation tradeoff implemented by our algorithm under different conditions (restricted vs. non-restricted number of classes), loss measures ( $\ell_{a,c}$ ,  $\ell_{rank,t}$ , and Hamming loss) and model/parameter settings ( $L$  = square loss,  $L$  = logistic loss, with varying  $R$ ).

**Datasets.** We used two multilabel datasets. The first one, called Mediamill, was introduced in a video annotation challenge [26]. It comprises 30,993 training samples and 12,914 test ones. The number of features  $d$  is 120, and the number of classes  $K$  is 101. The second dataset is Sony CSL Paris [22], made up of 16,452 train samples and 16,519 test samples, each sample being described by  $d = 98$  features. The number of classes  $K$  is 632. In both cases, feature vectors have been normalized to unit L2 norm.

**Parameter setting and loss measures.** We used the algorithm in Figure 1 with two different loss functions, the square loss and the logistic loss, and varied the parameter  $R$  for the latter. The setting of the cost function  $c(i, s)$  depends on the task at hand, and for this preliminary experiments we decided to evaluate two possible settings only. The first one, denoted by “decreasing  $c$ ” is  $c(i, s) = \frac{s-i+1}{s}, i = 1, \dots, s$ , the second one, denoted by “constant  $c$ ”, is  $c(i, s) = 1$ , for all  $i$  and  $s$ . In all experiments, the  $a$  parameter was set to 0.5, so that  $\ell_{a,c}$  with constant  $c$  reduces to half the Hamming loss. In the decreasing  $c$  scenario, we evaluated the performance of the algorithm on the loss  $\ell_{a,c}$  that the algorithm is minimizing, but also its ability to produce meaningful (partial) rankings through  $\ell_{rank,t}$ . On the constant  $c$  setting, we evaluated the Hamming loss. As is typical of multilabel problems, the label *density*, i.e., the average fraction of labels associated with the examples, is quite small. For instance, on Mediamill this is 4,3%. Hence, it is clearly beneficial to impose an upper bound  $S$  on  $|\hat{Y}_t|$ . For the constant  $c$  and ranking loss experiments we tried out different values of  $S$ , and reported the final performance.

**Baseline.** As baseline, we considered a full information version of Algorithm 1 using the square loss, that receives after each prediction the full array of true labels  $Y_t$  for each sample. We call this algorithm OBR (Online Binary Relevance), because it is a natural online adaptation of the binary relevance algorithm, widely used as a baseline in the multilabel literature. Comparing to OBR stresses the effectiveness of the exploration/exploitation rule above and beyond the details of underlying generalized

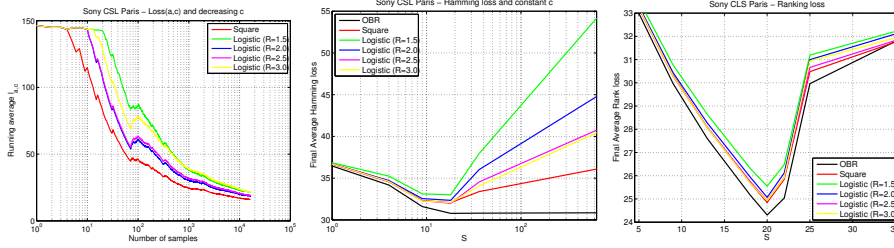


Figure 2: Experiments on the Sony CSL Paris dataset.

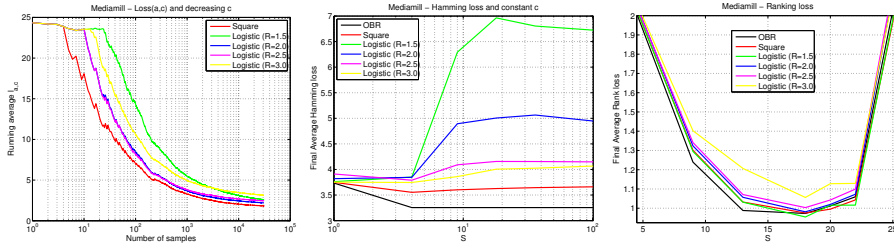


Figure 3: Experiments on the Mediamill dataset.

linear predictor. OBR was used to produce subsets (as in the Hamming loss case), and restricted rankings (as in the case of  $\ell_{rank,t}$ ).

**Results.** Our results are summarized in Figures 2 and 3. The algorithms have been trained by sweeping only once over the training data. Though preliminary in nature, these experiments allow us to draw a few conclusions. Our results for the average  $\ell_{a,c}(Y_t, \hat{Y}_t)$  with decreasing  $c$  are contained in the two left plots. We can see that the performance is improving over time on both datasets, as predicted by Theorem 2. In the middle plots are the final cumulative Hamming losses with constant  $c$  divided by the number of training samples, as a function of  $S$ . Similar plots are on the right with the final average ranking losses  $\ell_{rank,t}$ . In both cases we see that there is an optimal value of  $S$  that allows to balance the exploration and the exploitation of the algorithm. Moreover the performance of our algorithm is always pretty close to the performance of OBR, even if our algorithm is receiving only partial feedback. In many experiments the square loss seems to give better results. Exception is the ranking loss on the Mediamill dataset (Figure 3, right).

**Conclusions.** We have used generalized linear models to formalize the exploration-exploitation tradeoff in a multilabel/ranking setting with partial feedback, providing  $T^{1/2}$ -like regret bounds under semi-adversarial settings. Our analysis decouples the multilabel/ranking loss at hand from the label-generation model. Thanks to the usage of calibrated score values  $\hat{p}_{i,t}$ , our algorithm is capable of automatically inferring where to split the ranking between relevant and nonrelevant classes [11], the split being clearly induced by the loss parameters in  $\ell_{a,c}$ . We are planning on using more general label models that explicitly capture label correlations to be applied to other loss functions

(e.g., F-measure, 0/1, average precision, etc.). We are also planning on carrying out a more thorough experimental comparison, especially to full information multilabel methods that take such correlations into account. Finally, we are currently working on extending our framework to structured output tasks, like (multilabel) hierarchical classification.

## 6 Appendix

This appendix contains the proofs of all lemmas and theorems presented in the main text.

**Proof:** [Lemma 1] First observe that, for any given size  $s$ , the sequence  $Y_{s,t}^*$  must contain the  $s$  top-ranked classes in the sorted order of  $p_{i,t}$ . This is because, for any candidate sequence  $Y_s = \{j_1, j_2, \dots, j_s\}$ , we have  $\mathbb{E}_t[\ell_{a,c}(Y_t^*, Y_s)] = \sum_{i \in Y_s} \left( c(j_i, s) - \left( \frac{a}{1-a} + c(j_i, s) \right) p_{i,t} \right)$ . If there exists  $i \in Y_s$  which is not among the  $s$ -top ranked ones, then we could replace class  $i$  in position  $j_i$  within  $Y_s$  with class  $k \notin Y_s$  such that  $p_{k,t} > p_{i,t}$  obtaining a smaller loss.

Next, we show that the optimal ordering within  $Y_{s,t}^*$  is precisely ruled by the non-increasing order of  $p_{i,t}$ . By the sake of contradiction, assume there are  $i$  and  $k$  in  $Y_{s,t}^*$  such that  $i$  precedes  $k$  in  $Y_{s,t}^*$  but  $p_{k,t} > p_{i,t}$ . Specifically, let  $i$  be in position  $j_1$  and  $k$  be in position  $j_2$  with  $j_1 < j_2$  and such that  $c(j_1, s) > c(j_2, s)$ . Then switching the two classes within  $Y_{s,t}^*$  yields an expected loss difference of

$$\begin{aligned} & c(j_1, s) - \left( \frac{a}{1-a} + c(j_1, s) \right) p_{i,t} + c(j_2, s) - \left( \frac{a}{1-a} + c(j_2, s) \right) p_{k,t} \\ & - \left( c(j_1, s) - \left( \frac{a}{1-a} + c(j_1, s) \right) p_{k,t} \right) - \left( c(j_2, s) - \left( \frac{a}{1-a} + c(j_2, s) \right) p_{i,t} \right) \\ & = (p_{k,t} - p_{i,t}) (c(j_1, s) - c(j_2, s)) > 0. \end{aligned}$$

Hence switching would get a smaller loss which leads as a consequence to  $Y_{s,t}^* = (j_1, j_2, \dots, j_s)$ .  $\square$

The algorithm in Figure 1 works by updating through the gradients  $\nabla_{i,t}$  of a modular margin-based loss function  $\sum_{i=1}^K L(\mathbf{w}_i^\top \mathbf{x})$  associated with the label generation model (2) so as to make the parameters  $(\mathbf{u}_1, \dots, \mathbf{u}_K) \in \mathcal{R}^{dK}$  therein achieve the Bayes optimality condition

$$(\mathbf{u}_1, \dots, \mathbf{u}_K) = \arg \min_{\mathbf{w}_1, \dots, \mathbf{w}_K : \mathbf{w}_i^\top \mathbf{x}_t \in D} \mathbb{E}_t \left[ \sum_{i=1}^K L(s_{i,t} \mathbf{w}_i^\top \mathbf{x}_t) \right], \quad (4)$$

where  $\mathbb{E}_t[\cdot]$  above is over the generation of  $Y_t$  in producing the sign value  $s_{i,t} \in \{-1, 0, +1\}$ , conditioned on the past (in particular, conditioned on  $\hat{Y}_t$ ). The requirement in (4) is akin to the classical construction of *proper scoring rules* in the statistical literature (e.g., [23]).

The following lemma faces the problem of hand-crafting a convenient loss function  $L(\cdot)$  such that (4) holds.

**Lemma 5.** Let  $\mathbf{w}_1, \dots, \mathbf{w}_K \in \mathcal{R}^{dK}$  be arbitrary weight vectors such that  $\mathbf{w}_i^\top \mathbf{x}_t \in D$ ,  $i \in [K]$ ,  $(\mathbf{u}_1, \dots, \mathbf{u}_K) \in \mathcal{R}^{dK}$  be defined in (2),  $s_{i,t}$  be the updating signs computed by the algorithm at the end (Step 5) of time  $t$ ,  $L : D = [-R, R] \subseteq \mathcal{R} \rightarrow \mathcal{R}^+$  be a convex and differentiable function of its argument, with  $g(\Delta) = -L'(\Delta)$ . Then for any  $t$  we have

$$\mathbb{E}_t \left[ \sum_{i=1}^K L(s_{i,t} \mathbf{w}_i^\top \mathbf{x}_t) \right] \geq \mathbb{E}_t \left[ \sum_{i=1}^K L(s_{i,t} \mathbf{u}_i^\top \mathbf{x}_t) \right],$$

i.e., (4) holds.

**Proof:** Let us introduce the shorthands  $\Delta_i = \mathbf{u}_i^\top \mathbf{x}_t$ ,  $\hat{\Delta}_i = \mathbf{w}_i^\top \mathbf{x}_t$ ,  $s_i = s_{i,t}$ , and  $p_i = \mathbb{P}(y_{i,t} = 1 | \mathbf{x}_t) = \frac{L'(-\Delta_i)}{L'(\Delta_i) + L'(-\Delta_i)}$ . Moreover, let  $\mathbb{P}_t(\cdot)$  be an abbreviation for the conditional probability  $\mathbb{P}(\cdot | (y_1, \mathbf{x}_1), \dots, (y_{t-1}, \mathbf{x}_{t-1}), \mathbf{x}_t)$ . Recalling the way  $s_{i,t}$  is constructed (Figure 1), we can write

$$\begin{aligned} \mathbb{E}_t \left[ \sum_{i=1}^K L(s_{i,t} \hat{\Delta}_i) \right] &= \sum_{i \in \hat{Y}_t} \left( \mathbb{P}_t(s_{i,t} = 1) L(\hat{\Delta}_i) + \mathbb{P}_t(s_{i,t} = -1) L(-\hat{\Delta}_i) \right) + (K - |\hat{Y}_t|) L(0) \\ &= \sum_{i \in \hat{Y}_t} \left( p_i L(\hat{\Delta}_i) + (1 - p_i) L(-\hat{\Delta}_i) \right) + (K - |\hat{Y}_t|) L(0), \end{aligned}$$

For similar reasons,

$$\mathbb{E}_t \left[ \sum_{i=1}^K L(s_{i,t} \Delta_i) \right] = \sum_{i \in \hat{Y}_t} \left( p_i L(\Delta_i) + (1 - p_i) L(-\Delta_i) \right) + (K - |\hat{Y}_t|) L(0).$$

Since  $L(\cdot)$  is convex, so is  $\mathbb{E}_t \left[ \sum_{i=1}^K L(s_{i,t} \hat{\Delta}_i) \right]$  when viewed as a function of the  $\hat{\Delta}_i$ .

We have that  $\frac{\partial \mathbb{E}_t \left[ \sum_{i=1}^K L(s_{i,t} \hat{\Delta}_i) \right]}{\partial \hat{\Delta}_i} = 0$  if and only if for all  $i \in \hat{Y}_t$  we have that  $\hat{\Delta}_i$  satisfies

$$p_i = \frac{L'(-\hat{\Delta}_i)}{L'(\hat{\Delta}_i) + L'(-\hat{\Delta}_i)}.$$

Since  $p_i = \frac{L'(-\Delta_i)}{L'(\Delta_i) + L'(-\Delta_i)}$ , we have that  $\mathbb{E}_t \left[ \sum_{i=1}^K L(s_{i,t} \hat{\Delta}_i) \right]$  is minimized when  $\hat{\Delta}_i = \Delta_i$  for all  $i \in [K]$ . The claimed result immediately follows.  $\square$

Let now  $\text{Var}_t(\cdot)$  be a shorthand for  $\text{Var}(\cdot | (y_1, \mathbf{x}_1), \dots, (y_{t-1}, \mathbf{x}_{t-1}), \mathbf{x}_t)$ . The following lemma shows that under additional assumptions on the loss  $L(\cdot)$ , we are afforded to bound the variance of a difference of losses  $L(\cdot)$  by the expectation of this difference. This will be key to proving the fast rates of convergence contained in the subsequent Lemma 9.

**Lemma 6.** Let  $(\mathbf{w}'_{1,t}, \dots, \mathbf{w}'_{K,t}) \in \mathcal{R}^{dK}$  be the weight vectors computed by the algorithm in Figure 1 at the beginning (Step 2) of time  $t$ ,  $s_{i,t}$  be the updating signs computed at the end (Step 5) of time  $t$ , and  $(\mathbf{u}_1, \dots, \mathbf{u}_K) \in \mathcal{R}^{dK}$  be the comparison vectors defined through (2). Let  $L : D = [-R, R] \subseteq \mathcal{R} \rightarrow \mathcal{R}^+$  be a  $C^2(D)$  convex function of

its argument, with  $g(\Delta) = -L'(\Delta)$  and such that there are positive constants  $c'_L$  and  $c''_L$  with  $(L'(\Delta))^2 \leq c'_L$  and  $L''(\Delta) \geq c''_L$  for all  $\Delta \in D$ . Then for any  $i \in \hat{Y}_t$

$$0 \leq \text{Var}_t \left( L(s_{i,t} \mathbf{x}_t^\top \mathbf{w}'_{i,t}) - L(s_{i,t} \mathbf{u}_i^\top \mathbf{x}_t) \right) \leq \frac{2c'_L}{c''_L} \mathbb{E}_t \left[ L(s_{i,t} \mathbf{x}_t^\top \mathbf{w}'_{i,t}) - L(s_{i,t} \mathbf{u}_i^\top \mathbf{x}_t) \right].$$

**Proof:** Let us introduce the shorthands  $\Delta_i = \mathbf{x}_t^\top \mathbf{u}_i$ ,  $\hat{\Delta}_i = \mathbf{x}_t^\top \mathbf{w}'_{i,t}$ ,  $s_i = s_{i,t}$ , and  $p_i = \mathbb{P}(y_{i,t} = 1 \mid \mathbf{x}_t) = \frac{L'(-\Delta_i)}{L'(\Delta_i) + L'(-\Delta_i)}$ . Then, for any  $i \in [K]$ ,

$$\text{Var}_t \left( L(s_{i,t} \mathbf{x}_t^\top \mathbf{w}'_{i,t}) - L(s_{i,t} \mathbf{u}_i^\top \mathbf{x}_t) \right) \leq \mathbb{E}_t \left( \left( L(s_i \hat{\Delta}_i) - L(s_i \Delta_i) \right)^2 \right) \leq c'_L (\hat{\Delta}_i - \Delta_i)^2. \quad (5)$$

Moreover, for any  $i \in \hat{Y}_t$  we can write

$$\begin{aligned} \mathbb{E}_t \left[ L(s_i \hat{\Delta}_i) - L(s_i \Delta_i) \right] &= p_i (L(\hat{\Delta}_i) - L(\Delta_i)) + (1 - p_i) (L(-\hat{\Delta}_i) - L(-\Delta_i)) \\ &\geq p_i \left( L'(\Delta_i)(\hat{\Delta}_i - \Delta_i) + \frac{c''_L}{2} (\hat{\Delta}_i - \Delta_i)^2 \right) \\ &\quad + (1 - p_i) \left( L'(-\Delta_i)(\Delta_i - \hat{\Delta}_i) + \frac{c''_L}{2} (\hat{\Delta}_i - \Delta_i)^2 \right) \\ &= p_i \frac{c''_L}{2} (\hat{\Delta}_i - \Delta_i)^2 + (1 - p_i) \frac{c''_L}{2} (\hat{\Delta}_i - \Delta_i)^2 \\ &= \frac{c''_L}{2} (\hat{\Delta}_i - \Delta_i)^2, \end{aligned} \quad (6)$$

where the second equality uses the definition of  $p_i$ . Combining (5) with (6) gives the desired bound.  $\square$

We continue by showing a one-step regret bound for our original loss  $\ell_{a,c}$ . The precise connection to loss  $L(\cdot)$  will be established with the help of a later lemma (Lemma 9).

**Lemma 7.** Let  $L : D = [-R, R] \subseteq \mathcal{R} \rightarrow \mathcal{R}^+$  be a convex, twice differentiable, and nonincreasing function of its argument. Let  $(\mathbf{u}_1, \dots, \mathbf{u}_K) \in \mathcal{R}^{dK}$  be defined in (2) with  $g(\Delta) = -L'(\Delta)$  for all  $\Delta \in D$ . Let also  $c_L$  be a positive constant such that

$$\frac{L'(\Delta) L''(-\Delta) + L''(\Delta) L'(-\Delta)}{(L'(\Delta) + L'(-\Delta))^2} \geq -c_L$$

holds for all  $\Delta \in D$ . Finally, let  $\Delta_{i,t}$  denote  $\mathbf{u}_i^\top \mathbf{x}_t$ , and  $\hat{\Delta}'_{i,t}$  denote  $\mathbf{x}_t^\top \mathbf{w}'_{i,t}$ , where  $\mathbf{w}'_{i,t}$  is the  $i$ -th weight vector computed by the algorithm at the beginning (Step 2) of time  $t$ . If time  $t$  is such that  $|\Delta_{i,t} - \hat{\Delta}'_{i,t}| \leq \epsilon_{i,t}$  for all  $i \in [K]$ , then

$$\mathbb{E}_t[\ell_{a,c}(Y_t, \hat{Y}_t)] - \mathbb{E}_t[\ell_{a,c}(Y_t, Y_t^*)] \leq 2(1-a)c_L \sum_{i \in \hat{Y}_t} \epsilon_{i,t}.$$

**Proof:** Introduce the shorthand notation  $p(\Delta) = \frac{g(-\Delta)}{g(\Delta)+g(-\Delta)}$ . We can write

$$\begin{aligned} \mathbb{E}_t[\ell_{a,c}(Y_t, \hat{Y}_t)] - \mathbb{E}_t[\ell_{a,c}(Y_t, Y_t^*)] \\ = (1-a) \sum_{i \in \hat{Y}_t} \left( c(\hat{j}_i, |\hat{Y}_t|) - \left( \frac{a}{1-a} + c(\hat{j}_i, |\hat{Y}_t|) \right) p(\Delta_{i,t}) \right) \\ - (1-a) \sum_{i \in Y_t^*} \left( c(j_i^*, |Y_t^*|) - \left( \frac{a}{1-a} + c(j_i^*, |Y_t^*|) \right) p(\Delta_{i,t}) \right), \end{aligned}$$

where  $\hat{j}_i$  denotes the position of class  $i$  in  $\hat{Y}_t$  and  $j_i^*$  is the position of class  $i$  in  $Y_t^*$ . Now,

$$p'(\Delta) = \frac{-g'(-\Delta)g(\Delta) - g'(\Delta)g(-\Delta)}{(g(\Delta) + g(-\Delta))^2} = \frac{-L'(\Delta)L''(-\Delta) - L'(-\Delta)L''(\Delta)}{(L'(\Delta) + L'(-\Delta))^2} \geq 0$$

since  $g(\Delta) = -L'(\Delta)$ , and  $L(\cdot)$  is convex and nonincreasing. Hence  $p(\Delta)$  is itself a nondecreasing function of  $\Delta$ . Moreover, the extra condition on  $L$  involving  $L'$  and  $L''$  is a Lipschitz condition on  $p(\Delta)$  via a uniform bound on  $p'(\Delta)$ . Hence, from  $|\Delta_{i,t} - \hat{\Delta}'_{i,t}| \leq \epsilon_{i,t}$  and the definition of  $\hat{Y}_t$  we can write

$$\begin{aligned} \mathbb{E}_t[\ell_{a,c}(Y_t, \hat{Y}_t)] - \mathbb{E}_t[\ell_{a,c}(Y_t, Y_t^*)] \\ \leq (1-a) \sum_{i \in \hat{Y}_t} \left( c(\hat{j}_i, |\hat{Y}_t|) - \left( \frac{a}{1-a} + c(\hat{j}_i, |\hat{Y}_t|) \right) p([\hat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \right) \\ - (1-a) \sum_{i \in Y_t^*} \left( c(j_i^*, |Y_t^*|) - \left( \frac{a}{1-a} + c(j_i^*, |Y_t^*|) \right) p([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) \right) \\ \leq (1-a) \sum_{i \in \hat{Y}_t} \left( c(\hat{j}_i, |\hat{Y}_t|) - \left( \frac{a}{1-a} + c(\hat{j}_i, |\hat{Y}_t|) \right) p([\hat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \right) \\ - (1-a) \sum_{i \in \hat{Y}_t} \left( c(\hat{j}_i, |\hat{Y}_t|) - \left( \frac{a}{1-a} + c(\hat{j}_i, |\hat{Y}_t|) \right) p([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) \right) \\ = (1-a) \sum_{i \in \hat{Y}_t} \left( c(\hat{j}_i, |\hat{Y}_t|) \left( p([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) - p([\hat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \right) \right) \\ \leq 2(1-a)c_L \sum_{i \in \hat{Y}_t} \epsilon_{i,t}, \end{aligned}$$

the last inequality deriving from  $c(i, s) \leq 1$  for all  $i \leq s \leq K$ , and

$$p([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) - p([\hat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \leq c_L ([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D - [\hat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \leq 2c_L \epsilon_{i,t}. \quad \square$$

Likewise, we provide a similar bound for the ranking loss.

**Lemma 8.** *Under the same assumptions and notation as in Lemma 7, let the Algorithm in Figure 1 be working with  $\alpha \rightarrow 1$  and strictly decreasing cost values  $c(i, s)$ . Let  $\mathbf{w}_{i,t}$  be the  $i$ -th weight vector computed by this algorithm at the beginning (Step 2) of time*

*t*. If this algorithm ranks classes as  $\hat{p}_{j_1,t} \geq \dots \geq \hat{p}_{j_{S_t},t} \geq 0$ , and time *t* is such that  $|\Delta_{i,t} - \hat{\Delta}'_{i,t}| \leq \epsilon_{i,t}$  for all  $i \in [K]$ , then

$$\begin{aligned} \mathbb{E}_t[\ell_{rank,t}(Y_t, (\hat{p}_{j_1,t}, \dots, \hat{p}_{j_{S_t},t}, 0, \dots, 0))] - \mathbb{E}_t[\ell_{rank,t}(Y_t, (p_{i_1,t}, \dots, p_{i_{S_t},t}, 0, \dots, 0))] \\ \leq 2 S_t c_L \sum_{i \in \hat{Y}_t} \epsilon_{i,t}, \end{aligned}$$

where the  $p_{i,t} = \mathbb{P}_t(y_{i,t} = 1 | \mathbf{x}_t)$  are sorted as  $p_{i_1,t} \geq \dots \geq p_{i_{S_t},t} \geq 0$ , and  $\hat{Y}_t = (j_1, j_2, \dots, j_{S_t})$ .

**Proof:** Recall the notation  $\mathbb{P}_t(\cdot) = \mathbb{P}(\cdot | \mathbf{x}_t)$ , and  $p_{i,t} = p(\Delta_{i,t}) = \frac{g(-\Delta_{i,t})}{g(\Delta_{i,t}) + g(-\Delta_{i,t})}$ . Following [8] (proof of Theorem 2 therein), one can see that for generic sequences  $(\hat{p}_{1,t}, \dots, \hat{p}_{K,t})$  and  $(p_{1,t}, \dots, p_{K,t})$  one has

$$\begin{aligned} \mathbb{E}_t[\ell_{rank}(Y_t, (\hat{p}_{1,t}, \dots, \hat{p}_{K,t}))] - \mathbb{E}_t[\ell_{rank}(Y_t, (p_{1,t}, \dots, p_{K,t}))] \\ = \sum_{i,j \in [K], i < j} (\hat{r}(i,j) - r(i,j) + \hat{r}(j,i) - r(j,i)), \end{aligned} \quad (7)$$

where

$$\begin{aligned} \hat{r}(i,j) &= \mathbb{P}_t(y_{i,t} > y_{j,t}) (\{\hat{p}_{i,t} < \hat{p}_{j,t}\} + \frac{1}{2} \{\hat{p}_{i,t} = \hat{p}_{j,t}\}) \\ r(i,j) &= \mathbb{P}_t(y_{i,t} > y_{j,t}) (\{p_{i,t} < p_{j,t}\} + \frac{1}{2} \{p_{i,t} = p_{j,t}\}) \end{aligned}$$

Since

$$\mathbb{P}_t(y_{i,t} > y_{j,t}) - \mathbb{P}_t(y_{j,t} > y_{i,t}) = \mathbb{P}_t(y_{i,t} = 1) - \mathbb{P}_t(y_{j,t} = 1) = p_{i,t} - p_{j,t},$$

a simple case analysis reveals that

$$\hat{r}(i,j) - r(i,j) + \hat{r}(j,i) - r(j,i) = \begin{cases} \frac{1}{2} (p_{i,t} - p_{j,t}) & \text{If } \hat{p}_{i,t} < \hat{p}_{j,t}, p_{i,t} = p_{j,t} \text{ or } \hat{p}_{i,t} = \hat{p}_{j,t}, p_{i,t} > p_{j,t} \\ \frac{1}{2} (p_{j,t} - p_{i,t}) & \text{If } \hat{p}_{i,t} = \hat{p}_{j,t}, p_{i,t} < p_{j,t} \text{ or } \hat{p}_{i,t} > \hat{p}_{j,t}, p_{i,t} = p_{j,t} \\ p_{i,t} - p_{j,t} & \text{If } \hat{p}_{i,t} < \hat{p}_{j,t}, p_{i,t} > p_{j,t} \\ p_{j,t} - p_{i,t} & \text{If } \hat{p}_{i,t} > \hat{p}_{j,t}, p_{i,t} < p_{j,t}, \end{cases}$$

which can be uniformly upper bounded by  $|p_{i,t} - \hat{p}_{i,t}| + |p_{j,t} - \hat{p}_{j,t}|$ .

We now specialize the above to the two sequences  $(\hat{p}_{j_1,t}, \dots, \hat{p}_{j_{S_t},t}, 0, \dots, 0)$  and  $(p_{i_1,t}, \dots, p_{i_{S_t},t}, 0, \dots, 0)$ , and use  $\ell_{rank,t}$  instead of  $\ell_{rank}$ . Setting  $\hat{Y}_t = \{j_1, j_2, \dots, j_{S_t}\}$



and  $Y_t^* = \{i_1, i_2, \dots, i_{S_t}\}$ , and proceeding as in Lemma 7 we can write

$$\begin{aligned}
& \mathbb{E}_t[\ell_{rank,t}(Y_t, (\hat{p}_{j_1,t}, \dots, \hat{p}_{j_{S_t},t})] - \mathbb{E}_t[\ell_{rank,t}(Y_t, (p_{i_1,t}, \dots, p_{i_{S_t},t}))] \\
& \leq \sum_{i,j \in \hat{Y}_t, i < j} |p_{i,t} - \hat{p}_{i,t}| + |p_{j,t} - \hat{p}_{j,t}| + \sum_{i \in \hat{Y}_t} \left( p([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) - p([\hat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \right) \\
& = (S_t - 1) \sum_{i \in \hat{Y}_t} |p_{i,t} - \hat{p}_{i,t}| + \sum_{i \in \hat{Y}_t} \left( p([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) - p([\hat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \right) \\
& = (S_t - 1) \sum_{i \in \hat{Y}_t} |p(\Delta_{i,t}) - p([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D)| + \sum_{i \in \hat{Y}_t} \left( p([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) - p([\hat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \right) \\
& \leq (S_t - 1) \sum_{i \in \hat{Y}_t} c_L |\Delta_{i,t} - [\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D| + \sum_{i \in \hat{Y}_t} \left( p([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) - p([\hat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \right) \\
& \leq (S_t - 1) \sum_{i \in \hat{Y}_t} c_L \left( |\Delta_{i,t} - \hat{\Delta}'_{i,t}| + \epsilon_{i,t} \right) + \sum_{i \in \hat{Y}_t} \left( p([\hat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) - p([\hat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \right) \\
& \leq 2 S_t c_L \sum_{i \in \hat{Y}_t} \epsilon_{i,t},
\end{aligned}$$

as claimed.  $\square$

**Lemma 9.** Let  $L : D = [-R, R] \subseteq \mathcal{R} \rightarrow \mathcal{R}^+$  be a  $C^2(D)$  convex and nonincreasing function of its argument,  $(\mathbf{u}_1, \dots, \mathbf{u}_K) \in \mathcal{R}^{dK}$  be defined in (2) with  $g(\Delta) = -L'(\Delta)$  for all  $\Delta \in D$ , and such that  $\|\mathbf{u}_i\| \leq U$  for all  $i \in [K]$ . Assume there are positive constants  $c'_L$  and  $c''_L$  with  $(L'(\Delta))^2 \leq c'_L$  and  $L''(\Delta) \geq c''_L$  for all  $\Delta \in D$ . With the notation introduced in Figure 1, we have that

$$(\mathbf{x}^\top \mathbf{w}'_{i,t} - \mathbf{u}_i^\top \mathbf{x})^2 \leq \mathbf{x}^\top A_{i,t-1}^{-1} \mathbf{x} \left( U^2 + \frac{d c'_L}{(c''_L)^2} \ln \left( 1 + \frac{t-1}{d} \right) + \frac{12}{c''_L} \left( \frac{c'_L}{c''_L} + 3L(-R) \right) \ln \frac{K(t+4)}{\delta} \right)$$

holds with probability at least  $1 - \delta$  for any  $\delta < 1/e$ , uniformly over  $i \in [K]$ ,  $t = 1, 2, \dots$ , and  $\mathbf{x} \in \mathcal{R}^d$ .

**Proof:** For any given class  $i$ , the time- $t$  update rule  $\mathbf{w}'_{i,t} \rightarrow \mathbf{w}_{i,t+1} \rightarrow \mathbf{w}'_{i,t+1}$  in Figure 1 allows us to start off from [12] (proof of Theorem 2 therein), from which one can extract the following inequality

$$\begin{aligned}
& d_{i,t-1}(\mathbf{u}_i, \mathbf{w}'_{i,t}) \\
& \leq U^2 + \frac{1}{(c''_L)^2} \sum_{k=1}^{t-1} r_{i,k} - \frac{2}{c''_L} \sum_{k=1}^{t-1} \left( \nabla_{i,k}^\top (\mathbf{w}'_{i,k} - \mathbf{u}_i) - \frac{c''_L}{2} (s_{i,k} \mathbf{x}_k^\top (\mathbf{w}'_{i,k} - \mathbf{u}_i))^2 \right),
\end{aligned} \tag{8}$$

where we set  $r_{i,k} = \nabla_{i,k}^\top A_{i,k}^{-1} \nabla_{i,k}$ . Using the lower bound on the second derivative

of  $L$  we have

$$\begin{aligned}
& L(s_{i,k} \mathbf{x}_k^\top \mathbf{w}'_{i,k}) - L(s_{i,k} \mathbf{u}_i^\top \mathbf{x}_k) \\
& \leq L'(s_{i,k} \mathbf{x}_k^\top \mathbf{w}'_{i,k})(s_{i,k} \mathbf{x}_k^\top \mathbf{w}'_{i,k} - s_{i,k} \mathbf{u}_i^\top \mathbf{x}_k) - \frac{c_L''}{2} (s_{i,k} \mathbf{x}_k^\top \mathbf{w}'_{i,k} - s_{i,k} \mathbf{u}_i^\top \mathbf{x}_k)^2 \\
& = \nabla_{i,k}^\top (\mathbf{w}'_{i,k} - \mathbf{u}_i) - \frac{c_L''}{2} (s_{i,k} \mathbf{x}_k^\top (\mathbf{w}'_{i,k} - \mathbf{u}_i))^2.
\end{aligned}$$

Plugging back into (8) yields

$$d_{i,t-1}(\mathbf{u}_i, \mathbf{w}'_{i,t}) \leq U^2 + \frac{1}{(c_L'')^2} \sum_{k=1}^{t-1} r_{i,k} - \frac{2}{c_L''} \sum_{k=1}^{t-1} (L(s_{i,k} \mathbf{x}_k^\top \mathbf{w}'_{i,k}) - L(s_{i,k} \mathbf{u}_i^\top \mathbf{x}_k)) \quad (9)$$

We now borrow a proof technique from [5] (see also [7, 1] and references therein). Define  $L_{i,k} = L(s_{i,k} \mathbf{x}_k^\top \mathbf{w}'_{i,k}) - L(s_{i,k} \mathbf{u}_i^\top \mathbf{x}_k)$  and  $L'_{i,k} = \mathbb{E}_k[L_{i,k}] - L_{i,k}$ . Notice that the sequence of random variables  $L'_{i,1}, L'_{i,2}, \dots$ , forms a martingale difference sequence such that, for any  $i \in \hat{Y}_k$ :

- i.  $\mathbb{E}_k[L_{i,k}] \geq 0$ , by Lemma 6;
- ii.  $|L'_{i,k}| \leq 2L(-R)$ , since  $L(\cdot)$  is nonincreasing over  $D$ , and  $s_{i,k} \mathbf{x}_k^\top \mathbf{w}'_{i,k}, s_{i,k} \mathbf{u}_i^\top \mathbf{x}_k \in D$ ;
- iii.  $\text{Var}_k(L'_{i,k}) = \text{Var}_k(L_{i,k}) \leq \frac{2c_L'}{c_L''} \mathbb{E}_k[L_{i,k}]$  (again, because of Lemma 6).

On the other hand, when  $i \notin \hat{Y}_k$  then  $s_{i,k} = 0$ , and the above three properties are trivially satisfied. Under the above conditions, we are in a position to apply any fast concentration result for bounded martingale difference sequences. For instance, setting for brevity  $B = B(t, \delta) = 3 \ln \frac{K(t+4)}{\delta}$ , [17] allows us derive the inequality

$$\sum_{k=1}^{t-1} \mathbb{E}_k[L_{i,k}] - \sum_{k=1}^{t-1} L_{i,k} \geq \max \left\{ \sqrt{\frac{8c_L'}{c_L''} B \sum_{k=1}^{t-1} \mathbb{E}_k[L_{i,k}]}, 6L(-R) B \right\},$$

that holds with probability at most  $\frac{\delta}{Kt(t+1)}$  for any  $t \geq 1$ . We use the inequality  $\sqrt{cb} \leq \frac{1}{2}(c+b)$  with  $c = \frac{4c_L'}{c_L''} B$ , and  $b = 2 \sum_{k=1}^{t-1} \mathbb{E}_k[L_{i,k}]$ , and simplify. This gives

$$-\sum_{k=1}^{t-1} L_{i,k} \leq \left( \frac{2c_L'}{c_L''} + 6L(-R) \right) B$$

with probability at least  $1 - \frac{\delta}{Kt(t+1)}$ . Using the Cauchy-Schwarz inequality

$$(\mathbf{x}^\top \mathbf{w}'_{i,t} - \mathbf{u}_i^\top \mathbf{x})^2 \leq \mathbf{x}^\top A_{i,t-1}^{-1} \mathbf{x} d_{i,t-1}(\mathbf{u}_i, \mathbf{w}'_{i,t})$$

holding for any  $\mathbf{x} \in \mathcal{R}^d$ , and replacing back into (9) allows us to conclude that

$$(\mathbf{x}^\top \mathbf{w}'_{i,t} - \mathbf{u}_i^\top \mathbf{x})^2 \leq \mathbf{x}^\top A_{i,t-1}^{-1} \mathbf{x} \left( U^2 + \frac{1}{(c'_L)'^2} \sum_{k=1}^{t-1} r_{i,k} + \frac{12}{c'_L} \left( \frac{c'_L}{c''_L} + 3L(-R) \right) \ln \frac{K(t+4)}{\delta} \right) \quad (10)$$

holds with probability at least  $1 - \frac{\delta}{Kt(t+1)}$ , *uniformly* over  $\mathbf{x} \in \mathcal{R}^d$ .

The bounds on  $\sum_{k=1}^{t-1} r_{i,k}$  can be obtained in a standard way. Applying known inequalities (e.g., [3, 4, 12, 7]), and using the fact that  $\nabla_{i,k} = L'(s_{i,k} \mathbf{x}_k^\top \mathbf{w}'_{i,k}) s_{i,k} \mathbf{x}_k$  we have

$$\begin{aligned} \sum_{k=1}^{t-1} r_{i,k} &= \sum_{k=1}^{t-1} |s_{i,k}| (L'(s_{i,k} \mathbf{x}_k^\top \mathbf{w}'_{i,k}))^2 \mathbf{x}_k^\top A_{i,k}^{-1} \mathbf{x}_k \\ &\leq c'_L \sum_{k=1}^{t-1} |s_{i,k}| \mathbf{x}_k^\top A_{i,k}^{-1} \mathbf{x}_k \\ &\leq c'_L \sum_{k=1}^{t-1} \ln \frac{|A_{i,k}|}{|A_{i,k-1}|} \\ &= c'_L \ln \frac{|A_{i,t-1}|}{|A_{i,0}|} \\ &\leq d c'_L \ln \left( 1 + \frac{t-1}{d} \right). \end{aligned}$$

Piecing together as in (10) and stratifying over  $t = 1, 2, \dots$ , and  $i \in [K]$  concludes the proof.  $\square$

We are now ready to put all pieces together.

**Proof:** [Theorem 2] From Lemma 7 and Lemma 9, we see that with probability at least  $1 - \delta$ ,

$$R_T \leq 2(1-a) c_L \sum_{t=1}^T \sum_{i \in \hat{Y}_t} \epsilon_{i,t}, \quad (11)$$

when  $\epsilon_{i,t}^2$  is the one given in Figure 1. We continue by proving a pointwise upper bound on the sum in the RHS. More in detail, we will find an upper bound on  $\sum_{t=1}^T \sum_{i \in \hat{Y}_t} \epsilon_{i,t}^2$ , and then derive a resulting upper bound on the RHS of (11).

From Lemma 9 and the update rule (Step 5) of the algorithm we can write

$$\begin{aligned}
\epsilon_{i,t}^2 &\leq C \mathbf{x}_t^\top A_{i,t-1}^{-1} \mathbf{x}_t \\
&= C \frac{\mathbf{x}_t^\top (A_{i,t-1} + |s_{i,t}| \mathbf{x}_t \mathbf{x}_t^\top)^{-1} \mathbf{x}_t}{1 - |s_{i,t}| \mathbf{x}_t^\top (A_{i,t-1} + |s_{i,t}| \mathbf{x}_t \mathbf{x}_t^\top)^{-1} \mathbf{x}_t} \\
&= C \frac{\mathbf{x}_t^\top A_{i,t}^{-1} \mathbf{x}_t}{1 - |s_{i,t}| \mathbf{x}_t^\top (A_{i,t-1} + |s_{i,t}| \mathbf{x}_t \mathbf{x}_t^\top)^{-1} \mathbf{x}_t} \\
&\leq C \frac{\mathbf{x}_t^\top A_{i,t}^{-1} \mathbf{x}_t}{1 - |s_{i,t}| \mathbf{x}_t^\top (A_0 + |s_{i,t}| \mathbf{x}_t \mathbf{x}_t^\top)^{-1} \mathbf{x}_t} \\
&= C \frac{\mathbf{x}_t^\top A_{i,t}^{-1} \mathbf{x}_t}{1 - \frac{1}{2}} \\
&= 2C \mathbf{x}_t^\top A_{i,t}^{-1} \mathbf{x}_t.
\end{aligned}$$

Hence, if we set  $r_{i,t} = \mathbf{x}_t^\top A_{i,t}^{-1} \mathbf{x}_t$  and proceed as in the proof of Lemma 9, we end up with the upper bound  $\sum_{t=1}^T \epsilon_{i,t}^2 \leq 2Cd \ln(1 + \frac{T}{d})$ , holding for all  $i \in [K]$ . Denoting by  $M$  the quantity  $2Cd \ln(1 + \frac{T}{d})$ , we conclude from (11) that

$$R_T \leq 2(1-a)c_L \max \left\{ \sum_{i \in [K]} \sum_{t=1}^T \epsilon_{i,t} \mid \sum_{t=1}^T \epsilon_{i,t}^2 \leq M, \ i \in [K] \right\} = 2(1-a)c_L K \sqrt{TM},$$

as claimed.  $\square$

**Proof:** [Theorem 3] As we said, we change the definition of  $\epsilon_{i,t}^2$  in the Algorithm in Figure 1 to

$$\begin{aligned}
&\epsilon_{i,t}^2 = \\
&\max \left\{ \mathbf{x}^\top A_{i,t-1}^{-1} \mathbf{x} \left( \frac{2dc'_L}{(c''_L)^2} \ln \left( 1 + \frac{t-1}{d} \right) + \frac{12}{c''_L} \left( \frac{c'_L}{c''_L} + 3L(-R) \right) \ln \frac{K(t+4)}{\delta} \right), 4R^2 \right\}.
\end{aligned}$$

First, notice that the  $4R^2$  cap seamlessly applies, since  $(\mathbf{x}^\top \mathbf{w}'_{i,t} - \mathbf{u}_i^\top \mathbf{x})^2$  in Lemma 9 is bounded by  $4R^2$  anyway. With this modification, we have that Theorem 2 only holds for  $t$  such that  $\frac{dc'_L}{(c''_L)^2} \ln(1 + \frac{t-1}{d}) \geq U^2$ , i.e., for  $t \geq d \left( \exp \left( \frac{(c'_L)^2 U^2}{c'_L d} \right) - 1 \right) + 1$ , while for  $t < d \left( \exp \left( \frac{(c'_L)^2 U^2}{c'_L d} \right) - 1 \right) + 1$  we have in the worst-case scenario the maximum amount of regret at each step. From Lemma 7 we see that this maximum amount (the cap on  $\epsilon_{i,t}^2$  is needed here) can be bounded by  $4(1-a)c_L |\hat{Y}_t| R \leq 4(1-a)c_L K R$ .  $\square$

**Proof:** [Theorem 4] We start from the one step-regret delivered by Lemma 8, and

proceed as in the proof of Theorem 2. This yields

$$\begin{aligned}
R_T &\leq 2 c_L \sum_{t=1}^T S_t \sum_{i \in \hat{Y}_t} \epsilon_{i,t} \\
&\leq 2 S c_L \sum_{t=1}^T \sum_{i \in \hat{Y}_t} \epsilon_{i,t} \\
&\leq 2 S c_L \sum_{t=1}^T \sum_{i \in [K]} \epsilon_{i,t} \\
&= 2 S c_L \sum_{i \in [K]} \sum_{t=1}^T \epsilon_{i,t},
\end{aligned}$$

with probability at least  $1 - \delta$ , where  $\epsilon_{i,t}^2$  is the one given in Figure 1. Let  $M$  be as in the proof of Theorem 2. If  $N_{i,T}$  denotes the total number of times class  $i$  occurs in  $\hat{Y}_t$ , we have that  $\sum_{t=1}^T \epsilon_{i,t}^2 \leq M$ , implying  $\sum_{t=1}^T \epsilon_{i,t} \leq \sqrt{N_{i,T} M}$  for all  $i \in [K]$ . Moreover,  $\sum_{i \in [K]} N_{i,T} \leq ST$ . Hence

$$R_T \leq 2 S c_L \sum_{i \in [K]} \sqrt{N_{i,T} M} \leq 2 c_L \sqrt{M S K T},$$

as claimed.  $\square$

## References

- [1] Y. Abbasi-Yadkori, D. Pal, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *25th NIPS*, 2011.
- [2] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *JMLR*, 3, 2003.
- [3] K. S. Azoury and M. K. Warmuth. Relative loss bounds for online density estimation with the exponential family of distributions. *Machine Learning*, 43, 2001.
- [4] N. Cesa-Bianchi, A. Conconi, and C. Gentile. A second-order perceptron algorithm. *SIAM Journal on Computing*, 43:640–668, 2005.
- [5] K. Crammer and C. Gentile. Multiclass classification with bandit feedback using adaptive regularization. In *28th ICML*, 2011.
- [6] V. Dani, T. Hayes, and S. Kakade. Stochastic linear optimization under bandit feedback. In *21th Colt*, 2008.
- [7] O. Dekel, C. Gentile, and K. Sridharan. Robust selective sampling from single and multiple teachers. In *23rd Colt*, 2010.

- [8] K. Dembczynski, W. Waegeman, W. Cheng, and E. Hullermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, to appear.
- [9] S. Filippi, O. Cappé, A. Garivier, and C. Szepesvári. Parametric bandits: The generalized linear case. In *NIPS*, pages 586–594, 2010.
- [10] Y. Freund, R. D. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *JMLR*, 4:933–969, 2003.
- [11] J. Furnkranz, E. Hullermeier, E. Loza Menca, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73:133–153, 2008.
- [12] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.
- [13] E. Hazan and S. Kale. Online submodular minimization. In *NIPS 22*, 2009.
- [14] E. Hazan and S. Kale. Newtron: an efficient bandit algorithm for online multiclass prediction. In *NIPS*, 2011.
- [15] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*. MIT Press, 2000.
- [16] S. Kakade, S. Shalev-Shwartz, and A. Tewari. Efficient bandit algorithms for online multiclass prediction. In *25th ICML*, 2008.
- [17] S. Kakade and A. Tewari. On the generalization ability of online strongly convex programming algorithm. In *Nips*, 2008.
- [18] S. Kale, L. Reyzin, and R. Schapire. Non-stochastic bandit slate problems. In *24th NIPS*, 2010.
- [19] A. Krause and C. S. Ong. Contextual gaussian process bandit optimization. In *25th NIPS*, 2011.
- [20] T. H. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math*, 6, 1985.
- [21] P. McCullagh and J.A. Nelder. *Generalized linear models*. Chapman and Hall, 1989.
- [22] F. Pachet and P. Roy. Improving multilabel analysis of music titles: A large-scale validation of the correction approach. *IEEE Trans. on Audio, Speech, and Lang. Proc.*, 17(2):335–343, February 2009.
- [23] L.J. Savage. Elicitation of personal probabilities and expectations. *J. of the American Statistical Association*, 336:783–801, 1973.
- [24] P. Shivaswamy and T. Joachims. Online structured prediction via coactive learning. In *29th ICML, 2012*, to appear.

- [25] A. Slivkins, F. Radlinski, and S. Gollapudi. Learning optimally diverse rankings over large document collections. In *27th ICML*, 2010.
- [26] C. G. M. Snoek, M. Worring, J.C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proc. of the 14th ACM international conference on Multimedia*, MULTIMEDIA '06, pages 421–430, New York, NY, USA, 2006.
- [27] M. Streeter, D. Golovin, and A. Krause. Online learning of assignments. In *23rd NIPS*, 2009.
- [28] G. Tsoumakas, I. Katakis, and I. Vlahavas. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23:1079–1089, 2011.